An Event-Triggered ADP Control Approach for Continuous-Time System With Unknown Internal States

Xiangnan Zhong and Haibo He, *Senior Member, IEEE*

***Abstract*—This paper proposes a novel event-triggered adaptive dynamic programming (ADP) control method for nonlinear continuous-time system with unknown internal states. Comparing with the traditional ADP design with a fixed sample period, the event-triggered method samples the state and updates the controller only when it is necessary. Therefore, the computation cost and transmission load are reduced. Usually, the event-triggered method is based on the system entire state which is either infeasible or very difficult to obtain in practice applications. This paper integrates a neural-network-based observer to recover the system internal states from the measurable feedback. Both the proposed observer and the controller are aperiodically updated according to the designed triggering condition. Neural network techniques are applied to estimate the performance index and help calculate the control action. The stability analysis of the proposed method is also demonstrated by Lyapunov construct for both the continuous and jump dynamics. The simulation results verify the theoretical analysis and justify the efficiency of the proposed method.**

***Index Terms*—Adaptive dynamic programming (ADP), event-trigger, neural network, observer, online learning and control.**

I. Introduction

ADAPTIVE dynamic programming (ADP) has been studied and adopted for solving the Hamilton–Jacobi–Bellman (HJB) equation [1]–[4] in recent years. It has been widely recognized as one of the "core methodologies" to achieve the optimal control for intelligent systems in a general case [5]–[8]. Extensive efforts have been dedicated to developing ADP method from both the theoretical researches and real-world applications [9], [10]–[12]. A policy iteration ADP algorithm was developed for the discrete-time nonlinear system with stability analysis in [13]. A new performance index was established in [14] to solve the infinite-horizon optimal control problems of continuous-time complex-valued nonlinear systems. In [15] and [16], the systems with control constrains were considered and the corresponding control designs were developed based on ADP. Then, the ADP-based control scheme was also introduced into the robust control problems [17]–[19] to fill up the gap that dynamics uncertainties were not addressed. So far, most ADP control designs are based on entire state measurements in [20]–[22]. This is because ADP design needs to carefully evaluate the costs and benefits of the immediate action, as well as the choices which may acted in the future. If the system feedback are imperfect or unreliable indicators of the underlying process, this evaluation will become difficult [23]. However, in many real-world applications, the likelihood to access the complete knowledge of system state is either infeasible or very difficult to obtain. In other words, the feedback can only represent parts of the system states in these situations. In order to achieve better performance, estimating or reconstructing the state variables needs to be considered. Over the past decades, partially observable processes have attracted significantly increasing attention from both the artificial intelligence and machine learning areas. One major idea of most existing methods is to obtain the belief state, which is a sufficient statistic of the complete information of system and is also updated after each observation [24]–[26]. However, intensive computational burden will be caused when we try to obtain the belief state, especially when the dimension of the system state increases (i.e., curve of dimensionality). In these years, new iterative algorithms were developed under the partially observable environment based on reinforcement learning approach [27], [28]. Many of these methods, however, were still based on parameters/probability and required solid mathematic background to apply. Recently, ADP has been applied in this field and achieved some promising results. In [29], both the policy iteration and value iteration were provided using only the input–output data to obtain an optimal controller. This idea was extended on a linear tracking problem for unknown discrete-time system in [30]. Only the reduced information of the system dynamics was used in their method. In [31] and [32], an observer was established based on neural networks to determine a mapping between the behavior of the system and the external influences.

Because of the integration of an observer, the computation of ADP control design increases. Generally, the observer-based ADP methods rely on the periodic transmitted data with the fixed sampling period. This may bring huge number of the transmitted data and cause subsequently tremendous computation. This disadvantage becomes severe when the computation

Manuscript received October 6, 2015; revised January 21, 2016; accepted January 26, 2016. This work was supported in part by the National Science Foundation under Grant ECCS 1053717 and Grant IIS 1526835, and in part by the Army Research Office under Grant W911NF-12-1-0378. This paper was recommended by Associate Editor H. Zhang.

The authors are with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: xzhong@ele.uri.edu; he@ele.uri.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2016.2523878

bandwidth or sensor power sources are constrained. In recent years, the event-triggered control method [33]–[36] is introduced in ADP design. Different from the traditional method, the event-triggered method only transmits the system data and updates the control law when a specific event is triggered. In this way, the transmission load and computation burden are significantly reduced. Vamvoudakis in [37] online solved an event-triggered controller for a nonlinear system with guaranteed performance and without any linearizing process. In [38], a near optimal event-triggered condition of a nonlinear discrete-time system in affine form was provided. The authors extended this idea on the multi-input multi-output continuous-time system in [39] and provided the corresponding neural-network-based event-triggered condition.

In this paper, inspired by the above observations and literature studies, we propose a novel event-triggered ADP control method for the nonlinear continuous-time system with unknown internal states. The triggering condition is designed to make sure the control stability with the reduced information. Then, a neural-network-based observer is developed to recover the entire state from the system feedback. A critic network is established to approximate the performance index and help calculate the control law. Note that, in this paper, both the observer and the control law are updated according to the triggering condition. This means the observer and the control law are updated only when an specific event is triggered and held constant otherwise. The stability analysis for the close-loop system is presented using the Lyapunov construct for both the continuous and the jump dynamics. The major contributions of this paper include the following.

1) A neural-network-based observer is established and aperiodically updated with guaranteed stability to reconstruct the system internal states.
2) A triggering condition is designed for nonlinear continuous-time system only using the input–output measurements.
3) The event-triggered controller is designed based on the ADP technique.
4) The stability analysis for the close-loop system is explicitly provided for both the continuous and the jump dynamics.
5) The learning process for the observer used in this method is online rather than offline.

Comparing with [40], our proposed method only uses the triggered samples to update the observer and the control law, which reduces the transmission load and computation burden. Comparing with the works in [31] and [32], the proposed method can recover the details of what actually happened inside the partially observable dynamic processes.

The rest of this paper is organized as follows. In Section II, we formulate the problem of the event-triggered ADP method for partially observable nonlinear continuous-time system. The major results of this paper is provided in Section III, including four parts. First, an event-triggered regulator is designed based on the system input/output data with the corresponding stability proof. Then, a neural-network-based observer is established to recover the entire state from the reduced feedback. The designed observer is only updated when an event is triggered. Furthermore, the event-triggered ADP control scheme is presented. A critic network is built to approximate the performance index and help calculate the control law. In order to save the transmission load and computation burden, the control law is only updated according to the triggering data. The stability analysis of the close-loop system is then provided for both the continuous and the jump dynamics. In Section IV, a single-link robot arm case with reduced state dynamics is considered to verify the proposed method and theoretical analysis. Finally, Section V concludes this paper.

II. Problem Statement

Consider the nonlinear continuous-time system given as

$$\begin{aligned}\dot{x}(t) &= f(x(t)) + g(x(t))u(t)\\ y(t) &= Cx(t)\end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector with the initial state $x(0) = x_0$, $u(t) \in \mathbb{R}^m$ is the control input vector, $y(t) \in \mathbb{R}^p$ is the output vector, $f(x(t)) : \mathbb{R}^n \to \mathbb{R}^n$ and $g(x(t)) : \mathbb{R}^n \to \mathbb{R}^m$ are the unknown continuous-time state functions, and $f(0) = 0$. Assume that $f + gu$ is Lipschitz continuous on a set $\Omega \subseteq \mathbb{R}^n$ containing the origin. $C \in \mathbb{R}^{p*n}$ is the known output matrix.

Generally, the digital communication network is used to connect system, sensor, controller, and actuator in practical applications. Consider the limitation of the computation bandwidth or sensor power sources, an aperiodic updating and transmission rule for control action and system states is designed in this paper. In order to achieve this goal, a sampled-data system is introduced, which is characterized by a monotonically increasing sequence of sampling instants $\{\delta_j\}_{j=0}^{\infty}$, where $\delta_j < \delta_{j+1}$ for $j = 0, 1, 2, \ldots, \infty$. The time δ_j denotes the jth consecutive sampling instant. The output of the sample-data system is a sequence of the sampled states which can be denoted as

$$\hat{x}_j = x\big(\delta_j\big). \tag{2}$$

For simplicity, we assume that the sampled-data system has zero task delay.

Assumption 1 [40]: The nonlinear continuous-time system described in (1) is controllable and observable. Here, the system output, $y(t)$, is considered measured.

Therefore, a stabilizing controller can be guaranteed to be designed due to the controllability and the internal state can be ensured to be estimated from output measurement because of the observability. The control objective is to determine a feedback control law $u(t) = \mu(x(t))$ which minimizes the following infinite-horizon performance index:

$$\begin{aligned}V(x_0) &= \int_0^{\infty} \big(y^T(\tau)Qy(\tau) + u^T(\tau)Ru(\tau)\big)d\tau\\ &= \int_0^{\infty} U(y(\tau), u(\tau))d\tau\end{aligned} \tag{3}$$

where $U(y(t), u(t)) = y^T(t)Qy(t) + u^T(t)Ru(t)$ is the utility function with $U(0, 0) = 0$. Note that Q and R are symmetric positive definite matrices with appropriate dimensions. Here, the state-feedback control law is designed as $u(t) = \mu(\hat{x}_j, t)$, which maps the sampled state, rather than the continuous state in literature, onto a control vector. Therefore, the control signal $\mu(\hat{x}_j, t)$ is a piecewise constant function and consists of the control sequence $\{\mu(\hat{x}_j)\}_{j=0}^{\infty}$. In particular, $\{\mu(\hat{x}_j)\}_{j=0}^{\infty}$ becomes a continuous-time signal $\mu(\hat{x}_j, t)$ through a zero-order hold.

Let us recall the performance index in the traditional ADP method (time-triggered case)

$$V(x_0) = \int_0^{\infty} U(Cx(\tau), \mu(x(\tau)))d\tau = \int_0^{t} U(Cx(\tau), \mu(x(\tau)))d\tau + V(x(t)). \quad (4)$$

If the performance index (4) is continuously differentiable, then after transformation, we obtain

$$\lim_{t\to 0}[V(x(t)) - V(x_0)]/t = -\lim_{t\to 0}\frac{1}{t}\int_0^{t} U(Cx(\tau), \mu(x(\tau)))d\tau. \quad (5)$$

Therefore, the infinitesimal version of (4) is provided as

$$V_x^{*T}(f(x(t)) + g(x(t))\mu(x(t))) + U(Cx(t), \mu(x(t))) = 0 \quad (6)$$

where $V_x^* = \partial V^*(x(t))/\partial x(t)$ is the partial derivatives of the optimal performance index $V^*(x(t))$ with respect to $x(t)$.

Assume that the minimum of the left-hand side of (6) exists and is unique [41]. Therefore, the optimal control $\mu^*(x(t))$ satisfies the first-order necessary condition, which is given by the gradient of (6) with respect to $\mu(x(t))$. Hence, the optimal control law for the time-triggered case can be described as

$$u^*(t) = \mu^*(x(t)) = -\frac{1}{2}R^{-1}g^T(x(t))V_x^*. \quad (7)$$

In our event-triggered control design, the controller is only updated when an event is triggered. This means the controller is designed based on the sampled state $\hat{x}_j$ instead of the current state $x(t)$. Therefore, we obtain the event-triggered control law as

$$u^*(t) = \mu^*\big(\hat{x}_j, t\big) = -\frac{1}{2}R^{-1}g^T\big(\hat{x}_j\big)V_{\hat{x}_j}^* \quad (8)$$

where $V_{\hat{x}_j}^* = \partial V^*(\hat{x}_j)/\partial \hat{x}_j$. Note that, we use $\mu(\hat{x}_j)$ to represent $\mu(\hat{x}_j, t)$ to simplify the expression in the following presentation. By applying event-triggered control law (8) into (6), we obtain the event-triggered HJB equation

$$H\big(x(t), \mu^*\big(\hat{x}_j\big), V_x^*\big) = V_x^{*T}\left(f(x(t)) - \frac{1}{2}g(x(t))g^T\big(\hat{x}_j\big)V_{\hat{x}_j}^*\right) + \frac{1}{4}V_{\hat{x}_j}^{*T}g\big(\hat{x}_j\big)g^T\big(\hat{x}_j\big)V_{\hat{x}_j}^* + x^T(t)C^TQCx(t). \quad (9)$$

By using the event-triggered ADP method, the transmission load and computation burden can be significantly relaxed. However, we can observe that the system internal states $x(t)$, $\hat{x}_j$ are used in (8) and (9) to calculate the event-triggered controller and HJB equation. Since the knowledge of the system functions is completely unknown and the measured output can only represent parts of the system internal states, the existing ADP methods cannot be applied directly in this situation. In the next section, we will propose an event-triggered ADP control method using only the system input–output data. Note that, in order to simplify the presentation, we omit the time index t in the following statement.

III. Event-Triggered Controller Design Using Only the Input–Output Data

In this section, an event-triggered ADP control scheme using only the system input–output data is provided for continuous-time system. The general architecture of the proposed control method is shown in Fig. 1. First, because of the unavailability of the system internal state vectors and the system functions, a neural-network-based observer is designed to reconstruct both the state vector x and the control coefficient function $g(x)$ through an online manner. Therefore, the proposed observer design relaxes the requirement of an explicit identifier for $g(x)$ or an action network for $\mu(x)$. Then, the ADP framework is applied to approximate the performance index and calculate the optimal control vector. The critic network is established to estimate the performance index and it is trained online with a corresponding error term minimized overtime. Moreover, it is important to note that a sampled-data system is introduced with a sequence of sampling instants $\{\delta_j\}_{j=0}^{\infty}$ for both the neural network observer and the controller. This means both the observer and the controller are updated only when an specific event is triggered. The corresponding triggering condition is also provided. Due to the limitation of the communication bandwith and sensor power sources, this can significantly reduce the huge number of the transmitted data and subsequently tremendous computation.

In the following part, we will explicitly present the event-triggered ADP design using only the system input and output data. Specifically, in the first section, the triggering condition is derived for the sampled-data system. The corresponding stability analysis is also provided. A neural-network-based observer is designed in the second section, so that the control scheme can be developed using only the input and the output data measured during the operation of the system. A proof is also provided in this section to guarantee the stability of the observer and the accuracy of its estimation during the continuous and the jump dynamics. In the third section, neural network techniques are used to implement the proposed method. The weights updating rules for the critic network are also provided. Finally, the stability of the close-loop system is demonstrated using the Lyapunov theory for both dynamics. It is proved that the system state and parameter estimations are bounded, even when the trigger occurs.

A. Event-Triggered Regulator Design

Note that, since the internal state is unknown, an observer is designed to recover the system state. Therefore, the sampled states should be described as

$$\hat{x}_j = \hat{x}\big(\delta_j\big) \quad (10)$$

where $\hat{x}(\delta_j)$ is the estimated state at the sampled instants.

Now, define the gap function for $\forall t \in [\delta_j, \delta_{j+1})$ as

$$e_{y_j}(t) = C\hat{x}_j - y(t) \quad (11)$$

which is the difference between the term $C\hat{x}_j$ and the current system output.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

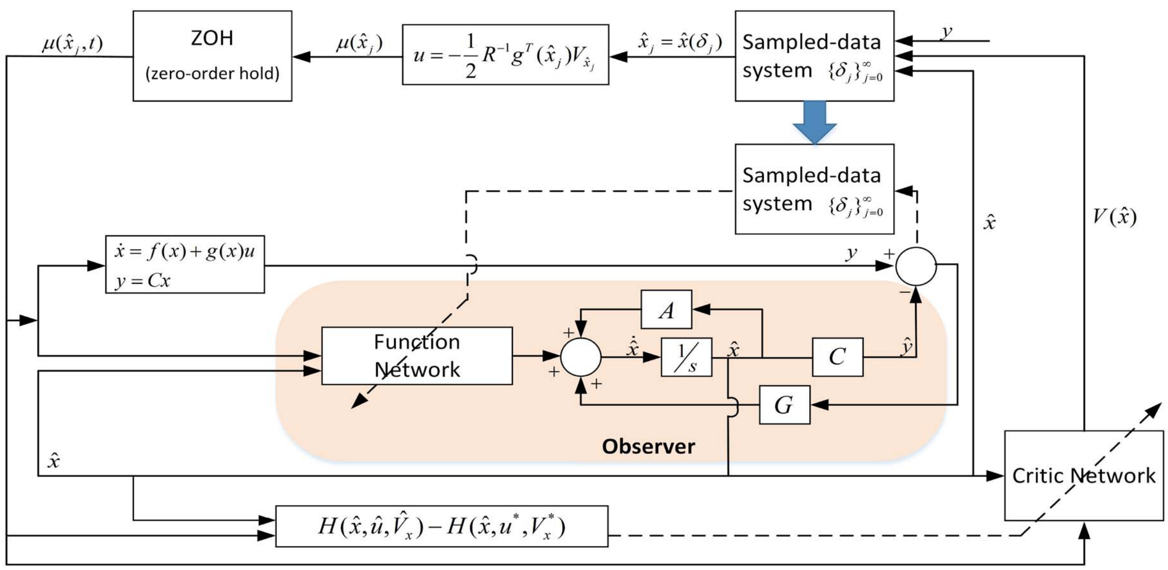


Fig. 1. Block diagram of the nonlinear continuous-time system control with only the input–output data.

Assumption 2: The controller is Lipschitz continuous with respect to the gap

$$\|\mu(x(t))-\mu(\hat{x}_j)\| \le L\|e_{x_j}\| \tag{12}$$

where L is a positive real constant and $e_{x_j}=\hat{x}_j-x(t)$.

Theorem 1: If there exists a positive definite function $V(x)$ that satisfies the HJB equation (9) with $V(0)=0$, and the control law is given in (8) with the triggering condition

$$\|e_{y_j}\|^2 \le \frac{(1-\alpha^2)\underline{\lambda}(Q)\|C\|^2\|y\|^2+\|C\|^2\|r^T\mu(x_j)\|^2}{L^2\|r\|^2} \tag{13}$$

then the close-loop system can be asymptotically stabilized, where $\alpha\in(0,1)$ is the designed parameter.

Proof: With the event-triggered control law (8), the orbital derivative of $V^*(x)$ along the system trajectory can be given as

$$\begin{aligned}\dot{V}^*(x) &= \left(\frac{\partial V^*(x)}{\partial x}\right)^T\dot{x}\\ &= V_x^*f(x)+V_x^{*T}g(x)\mu^*(\hat{x}_j).\end{aligned} \tag{14}$$

Here, consider the optimal control law and HJB equation in the traditional ADP method as

$$u^*=\mu^*(x)=-\frac{1}{2}R^{-1}g^T(x)V_x^* \tag{15}$$

and

$$V_x^{*T}f(x)-\frac{1}{4}V_x^{*T}g(x)R^{-1}g^T(x)V_x^*+y^TQy=0. \tag{16}$$

Therefore, we have

$$g^T(x)V_x^*=-2R\mu^*(x) \tag{17}$$

$$V_x^{*T}f(x)=\frac{1}{4}V_x^{*T}g(x)R^{-1}g^T(x)V_x^*-y^TQy. \tag{18}$$

Substitute (17) and (18) into (14), we obtain

$$\begin{aligned}\dot{V}^*(x) &= \frac{1}{4}V_x^{*T}g(x)R^{-1}g^T(x)V_x^*-y^TQy-2\mu^{*T}(x)R\mu^*(\hat{x}_j)\\ &= \mu^{*T}(x)R\mu^*(x)-2\mu^{*T}(x)R\mu^*(\hat{x}_j)-y^TQy.\end{aligned} \tag{19}$$

Since R is a symmetric positive definite matrix, we can describe R as $R=r\cdot r^T$. Therefore, we have

$$\begin{aligned}&\mu^{*T}(x)R\mu^*(x)-2\mu^{*T}(x)R\mu^*(\hat{x}_j)\\ &\quad=\|r^T\mu^*(x)-r^T\mu^*(\hat{x}_j)\|^2-\|r^T\mu^*(\hat{x}_j)\|^2.\end{aligned} \tag{20}$$

By using the Lipschitz condition in Assumption 2, we can write

$$\begin{aligned}\dot{V}^*(x) &= \|r^T\mu^*(x)-r^T\mu^*(\hat{x}_j)\|^2-\|r^T\mu^*(\hat{x}_j)\|^2-y^TQy\\ &\le -\|r^T\mu^*(\hat{x}_j)\|^2+L^2\|r\|^2\|e_{x_j}\|^2-\underline{\lambda}(Q)\|y\|^2\\ &= -\alpha^2\underline{\lambda}(Q)\|y\|^2+\Big[-(1-\alpha^2)\underline{\lambda}(Q)\|y\|^2\\ &\qquad+L^2\|r\|^2\|e_{x_j}\|^2-\|r^T\mu^*(\hat{x}_j)\|^2\Big].\end{aligned} \tag{21}$$

We know when the following inequality is satisfied:

$$\|e_{x_j}\|^2 \le \frac{(1-\alpha^2)\underline{\lambda}(Q)\|y\|^2+\|r^T\mu(\hat{x}_j)\|^2}{L^2\|r\|^2} \tag{22}$$

we have $\dot{V}^*(x)<0$.

Due to the unavailability of the current internal state, we obtain an equivalent condition (13) from (11). This is to say, when (13) is satisfied, we have $\dot{V}^*(x)<0$. Thus, in this way, $u^*=\mu^*(\hat{x}_j)$ can asymptotically stabilize the nonlinear continuous-time system (1). The conclusion holds. ■

It can be seen that the controller is guaranteed stable with the event-triggered sample data. The sampled-data system will continuously monitor the triggering condition (13). When a violation is about to occur, the sampled-data system will be triggered to sample the estimated system state, and according to the new sampled data, both the observer and the controller will be updated again.

B. Neural-Network-Based Observer Design

In this section, a neural-network-based observer is established to reconstruct the system state x and the control coefficient function $g(x)$. Consider system (1) with the event-triggered control law $\mu(\hat{x}_j)$. Choose a Hurwitz matrix A, such that the pair (C, A) is observable. The system dynamics (1) can be reformulated as

$$\begin{aligned}\dot{x} &= Ax + F_A(x) + g(x)\mu(\hat{x}_j)\\ y &= Cx\end{aligned} \tag{23}$$

where $F_A(x) = f(x) - Ax$. In order to reconstruct the state, we should identify the nonlinearity of the system. Since x is restricted to a compact set of $x \in \mathbb{R}^n$, the unknown system function can be described as a multilayer neural network with sufficiently large number of hidden layer neurons [42], then

$$\begin{aligned}&F_A(x) + g(x)\mu(\hat{x}_j)\\ &= \omega_{o2F}^{*T}\Phi_F(x) + \omega_{o2g}^{*T}\Phi_g(x)\mu(\hat{x}_j) + \varepsilon_F(x) + \varepsilon_g(x)\mu(\hat{x}_j)\\ &= \left[\omega_{o2F}^{*T}, \omega_{o2g}^{*T}\right]\begin{bmatrix}\Phi_F(x) & 0\\ 0 & \Phi_g(x)\end{bmatrix}\begin{bmatrix}1\\ \mu(\hat{x}_j)\end{bmatrix}\\ &\quad + [\varepsilon_F(x), \varepsilon_g(x)]\begin{bmatrix}1\\ \mu(\hat{x}_j)\end{bmatrix}\\ &= \omega_{o2}^{*T}\Phi(x) + \varepsilon(x)\end{aligned} \tag{24}$$

where ω_{o2}^* is the ideal weights of the neural network output layer, $\|\varepsilon(x)\| \le \varepsilon_M$ is the bounded neural network approximation error, $\Phi(\cdot)$ is the bounded sigmoid function that can be expressed as

$$\|\Phi(\cdot)\| = \left\|\frac{1 - e^{-(\cdot)}}{1 + e^{-(\cdot)}}\right\| \le \Phi_M. \tag{25}$$

It is assumed that the ideal weights are bounded as $\|\omega_{o2}^*\| \le \omega_{o2M}$. Moreover, we have

$$\omega_{o2}^* = \left[\omega_{o2F}^*, \omega_{o2g}^*\right] \tag{26}$$

$$\Phi(x) = \begin{bmatrix}\Phi_F(x) & 0\\ 0 & \Phi_g(x)\end{bmatrix}\begin{bmatrix}1\\ \mu(\hat{x}_j)\end{bmatrix} \tag{27}$$

$$\varepsilon(x) = [\varepsilon_F(x), \varepsilon_g(x)]\begin{bmatrix}1\\ \mu(\hat{x}_j)\end{bmatrix}. \tag{28}$$

Hence, the system states can be identified by updating the corresponding neural network weights. Since the ideal weights ω_{o2}^* are unknown, we establish a neural network, which is called the function network in this paper, to identify the nonlinearity by using the current estimates $\hat{\omega}_{o2}$ of the ideal weights ω_{o2}^*

$$\hat{F}_A(\hat{x}) + \hat{g}(\hat{x})\mu(\hat{x}_j) = \hat{\omega}_{o2}^T\Phi(\hat{x}). \tag{29}$$

It is important to note that in order to save the resource, the function network weights are only updated when an event is triggered, that is

$$\hat{\omega}_{o2j} = \hat{\omega}_{o2}(\delta_j). \tag{30}$$

Then, (29) becomes

$$\hat{F}_A(\hat{x}) + \hat{g}(x)\mu(\hat{x}_j) = \hat{\omega}_{o2j}^T\Phi(\hat{x}_j). \tag{31}$$

Hence, we design the following neural-network-based observer which is assumed to be of the Luenberger like structure:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + \hat{\omega}_{o2j}^T\Phi(\hat{x}_j) + G(y - \hat{y})\\ \hat{y} &= C\hat{x}\end{aligned} \tag{32}$$

where $\hat{x}$ and $\hat{y}$ are the estimated state and output of the observer, respectively, $\hat{x}_j$ is the estimated sampled state, and $G \in \mathbb{R}^{n*m}$ is the observer gain. Here, $\Phi(\hat{x}_j) = \Phi(\omega_{o1}\hat{X}_{oj})$, in which $\hat{X}_{oj} = [\hat{x}_j, \mu(\hat{x}_j)]$ is the input of the function network, and ω_{o1} is the weights of the function network hidden layer. Now, define the state estimation error as

$$\begin{aligned}\dot{\tilde{x}} &= \dot{x} - \dot{\hat{x}}\\ &= Ax + \omega_{o2}^{*T}\Phi(x) - A\hat{x} - \hat{\omega}_{o2j}^T\Phi(\hat{x}_j)\\ &\quad - G(y - \tilde{y}) + \varepsilon(x).\end{aligned} \tag{33}$$

By adding and subtracting $\omega_{o2}^{*T}\Phi(\hat{x}_j)$ from (33), such error dynamics become

$$\dot{\tilde{x}} = A_c\tilde{x} + \tilde{\omega}_{o2j}^T\Phi(\hat{x}_j) + \xi(x) \tag{34}$$

where $\tilde{\omega}_{o2j} = \omega_{o2}^* - \hat{\omega}_{o2j}$ is the neural network estimation error, $A_c = A - GC$ is a Hurwitz matrix, and $\xi(x) = \omega_{o2}^{*T}[\Phi(x) - \Phi(\hat{x}_j)] + \varepsilon(x)$ is a bounded disturbance term. This means, $\|\xi(x)\| \le \xi_M$ for some positive constant, due to the boundedness of the sigmoid function and the ideal neural network weights ω_{o2}^*.

Note that, in this paper, the input-to-hidden layer weights ω_{o1} are randomly chosen and kept constantly during the training process. Therefore, our goal now should be to find the updating rule for the hidden-to-output layer weights $\hat{\omega}_{o2j}$. Adjusting the weights of the function network is to minimize the squared error

$$E_o = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}\tilde{y}^2 \tag{35}$$

where $\tilde{y} = y - \hat{y}$. Since the updating law for the observer will have an aperiodic nature, it has to be updated only at the trigger instants and held constant otherwise. We can describe the following updating laws: when an event is not triggered, we have

$$\dot{\hat{\omega}}_{o2j} = 0, \quad \text{for } \delta_{j-1} \le t < \delta_j \tag{36}$$

and when an event is triggered, the jump equation to calculate $\hat{\omega}_{o2j}$ is given by

$$\begin{aligned}\hat{\omega}_{o2j}^+ &= \hat{\omega}_{o2j} - \beta_o\frac{\partial E_o}{\partial\hat{\omega}_{o2j}} - \rho\|\tilde{y}\|\hat{\omega}_{o2j}\\ &= \hat{\omega}_{o2j} - \beta_o\frac{\partial E_o}{\partial\tilde{y}}\frac{\partial\tilde{y}}{\partial\hat{x}}\frac{\partial\hat{x}}{\partial\hat{\omega}_{o2j}} - \rho\|\tilde{y}\|\hat{\omega}_{o2j} \quad \text{for } t = \delta_j\end{aligned} \tag{37}$$

where $\beta_o > 0$ is the learning rate of the function network and $\rho > 0$ is a small positive number. Note that the second term in (37) is the backpropagation term and the third term is the e-modification term for incorporating damping. We have $(\partial E_o/\partial\tilde{y}) = \tilde{y}$ and $(\partial\tilde{y}/\partial\hat{x}) = C^T$ according to (35) and (23), respectively. The updating rule can thus be achieved for solving the gradient $(\partial\hat{x}/\partial\hat{\omega}_{o2j})$. To solve this problem, we apply the static approximation of the gradient by setting $\dot{\hat{x}} = 0$ in (32). Then, after transformation, we obtain

$$\frac{\partial\hat{x}}{\partial\omega_{o2j}} = -A_c^{-T}\Phi(\hat{x}_j). \tag{38}$$

Hence, we have the updating rule for the function network at the trigger instants as

$$\hat{\omega}_{o2j}^{+} = \hat{\omega}_{o2j} - \beta_o\left(\tilde{y}^T CA_c^{-1}\right)^T \Phi\left(\hat{x}_j\right) - \rho\|\tilde{y}\|\hat{\omega}_{o2j} \quad \text{for } t = \delta_j. \tag{39}$$

In order to guarantee the stability of the neural-network-based observer and the accuracy of the estimation, the boundedness of the observer error should be provided for both the continuous and the jump dynamics.

Theorem 2: Consider the nonlinear continuous-time system given by (1) with the event-triggered neural-network-based observer given by (32). If the tuning laws for the function network of the observer are provided as (36) and (39) for different time instants, then the state estimation error $\tilde{x}$ and weight estimation errors $\tilde{\omega}_{o2j} = \omega_{o2j}^* - \hat{\omega}_{o2j}$ are uniformly ultimately bounded (UUB).

Proof: Since the observer is updated only when the event is triggered, we have to consider both the continuous and the jump dynamics separately. Initially, we will consider the following Lyapunov function L_o:

$$L_o = \frac{1}{2}\tilde{x}^T P\tilde{x} + \frac{1}{2}tr\left(\tilde{\omega}_{o2j}^T\tilde{\omega}_{o2j}\right) \tag{40}$$

where $\tilde{x}$ is the state estimation error given by (34) and $\tilde{\omega}_{o2j}$ is the weight estimation error. P is a positive definite matrix that satisfies

$$A_c^T P + PA_c = -M \tag{41}$$

where M is a positive definite matrix.

For the continuous dynamics of the observer model, by taking the time derivative of (40) with respect to the close-loop system trajectories, the second term has a zero derivative due to the function network continuous dynamics (36). Therefore

$$\begin{aligned}\dot{L}_o &= \frac{1}{2}\dot{\tilde{x}}^T P\tilde{x} + \frac{1}{2}\tilde{x}^T P\dot{\tilde{x}}\\ &= \frac{1}{2}\left(A_c\tilde{x} + \tilde{\omega}_{o2j}^T\Phi\left(\hat{x}_j\right) + \xi(x)\right)^T P\tilde{x}\\ &\quad + \frac{1}{2}\tilde{x}^T P\left(A_c\tilde{x} + \tilde{\omega}_{o2j}^T\Phi\left(\hat{x}_j\right) + \xi(x)\right).\end{aligned} \tag{42}$$

By using some polynomial adjustments and (41), (42) can be rewritten as

$$\begin{aligned}\dot{L}_o &= -\frac{1}{2}\tilde{x}^T M\tilde{x} + \tilde{x}^T P\left(\tilde{\omega}_{o2j}^T\Phi\left(\hat{x}_j\right) + \xi(x)\right)\\ &\le -\frac{1}{2}\underline{\lambda}(M)\|\tilde{x}\|^2 + \|\tilde{x}\|\|P\|\left(\|\tilde{\omega}_{o2}\|\Phi_M + \xi_M\right)\\ &\le -\frac{1}{2}\underline{\lambda}(M)\|\tilde{x}\|^2 + \left(2\omega_{oM}\Phi_M\|P\| + \|P\|\xi_M\right)\|\tilde{x}\|\end{aligned} \tag{43}$$

where $\underline{\lambda}(M)$ is the minimal eigenvalue of M. Hence, in order to guarantee the negativeness of the time derivative $\dot{L}_o$ at the continuous dynamics, the following condition on the state estimation error should hold:

$$\begin{aligned}\|\tilde{x}\| &\ge \frac{4\omega_M\Phi_M\|P\| + 2\xi_M\|P\|}{\underline{\lambda}(M)}\\ &= d.\end{aligned} \tag{44}$$

According to the Lyapunov extension theorem, as long as condition (44) is satisfied, it demonstrates that the state and the weights estimation errors are UUB.

Note that $\dot{L}_o$ for continuous dynamics is negative definite under the condition (44), which means $\tilde{x}$ is UUB outside the ball with radius d described as $X = \{\tilde{x}|\|\tilde{x}\| > d\}$. The size of the estimation error bound d can be kept arbitrarily small by proper selection of the parameters.

Next, we have to consider the jump dynamics. The function network weights are updated at these instants. For that reason, we consider the following form:

$$\begin{aligned}\Delta L_o &= \frac{1}{2}\left(\tilde{x}^T\right)^+ P\tilde{x}^+ - \frac{1}{2}\tilde{x}^T P\tilde{x}\\ &\quad + \frac{1}{2}tr\left(\left(\tilde{\omega}_{o2j}^T\right)^+\tilde{\omega}_{o2j}^+\right) - \frac{1}{2}tr\left(\tilde{\omega}_{o2j}^T\tilde{\omega}_{o2j}\right), \quad t = \delta_j.\end{aligned} \tag{45}$$

Since we have proved that the state estimation error is asymptotically stable, there exists

$$\frac{1}{2}\left(\tilde{x}^T\right)^+ P\tilde{x}^+ \le \frac{1}{2}\tilde{x}^T P\tilde{x}. \tag{46}$$

Therefore, the problem becomes to find a bound for the following term:

$$\Delta L_{o1}\left(\tilde{\omega}_{o2j}\right) = \frac{1}{2}tr\left(\left(\tilde{\omega}_{o2j}^T\right)^+\tilde{\omega}_{o2j}^+\right) - \frac{1}{2}tr\left(\tilde{\omega}_{o2j}^T\tilde{\omega}_{o2j}\right), \quad t = \delta_j. \tag{47}$$

Consider (39), we obtain

$$\begin{aligned}\tilde{\omega}_{o2j}^+ &= \omega_{o2j}^* - \hat{\omega}_{o2j}^+\\ &= \tilde{\omega}_{o2j} + \beta_o\left(\tilde{y}^T CA_c^{-1}\right)^T\Phi\left(\hat{x}_j\right) + \rho\|\tilde{y}\|\hat{\omega}_{o2j}.\end{aligned} \tag{48}$$

Substituting (48) into (47) and after some mathematical manipulation, the first difference $\Delta L_{o1}(\tilde{\omega}_{o2j})$ becomes

$$\begin{aligned}\Delta L_{o1}\left(\tilde{\omega}_{o2j}\right) &= tr\left(\tilde{\omega}_{o2j}^T\left(\beta_o\left(\tilde{y}^T CA_c^{-1}\right)^T\Phi\left(\hat{x}_j\right) + \rho\|\tilde{y}\|\hat{\omega}_{o2j}\right)\right)\\ &\quad + \left\|\beta_o\left(\tilde{y}^T CA_c^{-1}\right)^T\Phi\left(\hat{x}_j\right) + \rho\|\tilde{y}\|\hat{\omega}_{o2j}\right\|^2\\ &= tr\left(\tilde{\omega}_{o2j}^T\beta_o A_c^{-T}C^T\tilde{y}\Phi\left(\hat{x}_j\right) + \rho\|\tilde{y}\|\tilde{\omega}_{o2j}^T\omega_{o2j}^*\right.\\ &\quad \left. - \rho\|\tilde{y}\|\tilde{\omega}_{o2j}^T\tilde{\omega}_{o2j}\right) + \left\|\beta_o A_c^{-T}C^T\tilde{y}\Phi\left(\hat{x}_j\right)\right\|^2\\ &\quad + 2\Phi^T\left(\hat{x}_j\right)\tilde{y}^T\left(\beta_o A_c^{-T}C^T\right)^T\\ &\quad \cdot\rho\|\tilde{y}\|\hat{\omega}_{o2j} + \rho^2\|\tilde{y}\|^2\hat{\omega}_{o2j}^T\hat{\omega}_{o2j}\\ &\le -\rho\|C\|\|\tilde{x}\|\left\|\tilde{\omega}_{o2j}\right\|^2 + \|m\|\|\tilde{x}\|\left\|\tilde{\omega}_{o2j}\right\|\Phi_M\\ &\quad + \rho\|C\|\|\tilde{x}\|\left\|\tilde{\omega}_{o2j}\right\|\omega_{oM} + \|m\|^2\|\tilde{x}\|^2\Phi_M^2\\ &\quad + 2\rho\|C\|\|m\|\|\tilde{x}\|^2\Phi_M\left\|\hat{\omega}_{o2j}\right\|\\ &\quad + \rho^2\|C\|^2\|\tilde{x}\|^2\omega_{oM}^2\end{aligned} \tag{49}$$

where $m = \beta_o A_c^{-T} C^T C$. By completing the square of $\|\tilde{\omega}_{o2j}\|$, (49) becomes

$$\begin{aligned}\Delta L_{o1}(\tilde{\omega}_{o2j}) \leq & -\frac{1}{2}\Big(\|m\|\Phi_M + \rho\|C\|\omega_M - \|\tilde{\omega}_{o2j}\|\Big)^2\|\tilde{x}\| \\ & - \left(\rho - \frac{1}{2}\right)\|\tilde{\omega}_{o2j}\|^2\|\tilde{x}\| \\ & + \frac{1}{2}\Big(\|m\|\Phi_M + \rho\|C\|\|\omega_{oM}\|\Big)^2\|\tilde{x}\| \\ & + \Big(\|m\|^2\Phi_M^2 + 2\rho\|m\|\|C\|\|\omega_M\|^2\Phi_M \\ & \quad + \rho^2\|C\|^2\omega_{oM}^2\Big)\|\tilde{x}\|^2. \end{aligned} \tag{50}$$

Since $\|\tilde{x}\|$ is guaranteed positive, then $\Delta L_o(\tilde{\omega}_{o2j}) \leq 0$ is equivalent to the following condition holding:

$$\begin{aligned} & -\frac{1}{2}\big(\|m\|\Phi_M + \rho\|C\|\omega_M - \|\tilde{\omega}_{o2j}\|\big)^2 - \left(\rho - \frac{1}{2}\right)\|\tilde{\omega}_{o2j}\|^2 \\ & + \frac{1}{2}(\|m\|\Phi_M + \rho\|C\|\|\omega_{oM}\|)^2 \\ & + \Big(\|m\|^2\Phi_M^2 + 2\rho\|m\|\|C\|\|\omega_M\|^2\Phi_M \\ & \quad + \rho^2\|C\|^2\omega_{oM}^2\Big)\|\tilde{x}\| \leq 0. \end{aligned} \tag{51}$$

By defining

$$\begin{aligned}\gamma^2 = & \frac{1}{2}(\|m\|\Phi_M + \rho\|C\|\|\omega_{oM}\|)^2 \\ & + \Big(\|m\|^2\Phi_M^2 + 2\rho\|m\|\|C\|\|\omega_M\|^2\Phi_M \\ & \quad + \rho^2\|C\|^2\omega_{oM}^2\Big)\|\tilde{x}\| \end{aligned} \tag{52}$$

condition (51) becomes

$$\begin{aligned} & -\frac{1}{2}\Big(\|m\|\Phi_M + \rho\|C\|\omega_M - \|\tilde{\omega}_{o2j}\|\Big)^2 \\ & \qquad - \left(\rho - \frac{1}{2}\right)\|\tilde{\omega}_{o2j}\|^2 + \gamma^2 \leq 0. \end{aligned} \tag{53}$$

Note that due to the boundedness of the state estimation error has been proved, there exists a bound for γ^2. Therefore, we can prove that the jump dynamics are UUB as long as the following conditions satisfied:

$$\rho > \frac{1}{2} \tag{54}$$

$$\|\tilde{\omega}_{o2j}\| \geq \sqrt{\frac{\gamma^2}{\left(\rho - \frac{1}{2}\right)}}. \tag{55}$$

Hence, the system states estimation error and the neural network weight estimation errors are UUB in both the continuous and the jump dynamics. This completes the proof. ■

C. Optimal Event-Triggered Control Scheme Design

Neural network technique is applied in this section to implement the proposed event-triggered ADP method. A critic network is built to approximate the performance index which can be formulated as

$$V^*(x) = \omega_{c2}^{*T}\Phi(m(x)) + \varepsilon_c(x) \tag{56}$$

where ω_{c2}^* is the optimal weights between the hidden and the output layer of the critic network, $m(x) = \omega_{c1}^{*T}X_c$ to which ω_{c1}^* is the optimal input-to-hidden layer weights, $X_c = [x^T, \mu^T(x)]^T$, and $\|\varepsilon_c(x)\| \leq \varepsilon_{cM}$ is the bounded critic network error.

According to (56), the performance index $V^*(x)$ in the event-triggered control scheme can be approximated as

$$\hat{V}(\hat{x}) = \hat{\omega}_{c2}^T\Phi(m(\hat{x})) \tag{57}$$

where $\hat{V}(\hat{x})$ represent the estimated performance index, $\hat{\omega}_{c2}^T$ is the approximated hidden-to-output layer weights of the critic network, and $m(\hat{x}) = \hat{\omega}_{c1}^T\hat{X}_c$ to which $\hat{\omega}_{c1}$ is the estimated input-to-hidden layer weights of critic network and $\hat{X}_c = [\hat{x}, \mu(\hat{x}_j)]$ is the input of the critic network. We fix the input-to-hidden layer weights as ω_{c1}, which are chosen randomly at initial. Therefore, only the hidden-to-output layer weights $\hat{\omega}_{c2}$ need to be updated.

Define the error function for the critic network as

$$\begin{aligned} e_c &= H\Big(\hat{x}, \mu(\hat{x}_j), \hat{V}_x\Big) - H\big(x, u^*, V_x^*\big)\Big) \\ &= \left(\left(\frac{\partial\Phi(m(\hat{x}))}{\partial\hat{x}}\right)^T\hat{\omega}_{c2}\right)^T\dot{\hat{x}} + U\big(\hat{x}, \mu(\hat{x}_j)\big). \end{aligned} \tag{58}$$

We know that $H(x, u^*, V_x^*) = 0$ from (9). Adjusting the weights of the critic network is to minimize the objective function

$$E_c = \frac{1}{2}e_c^2. \tag{59}$$

Therefore, the hidden-to-output layer weights of the critic network can be updated as

$$\begin{aligned}\dot{\hat{\omega}}_{c2} &= -\beta_c\frac{\partial E_c}{\partial\hat{\omega}_{c2}} = -\beta_c\frac{\partial E_c}{\partial e_c}\frac{\partial e_c}{\partial\hat{\omega}_{c2}} \\ &= -\beta_c\frac{\kappa}{(\kappa^T\kappa + 1)^2}\big(\hat{\omega}_{c2}^T\kappa + U(\hat{x}, \mu(\hat{x}_j))\big)^2 \end{aligned} \tag{60}$$

where $\kappa = ((\partial\Phi(m(\hat{x}))/\partial\hat{x}))^T\dot{\hat{x}}$ and $\beta_c > 0$ is the learning rate of the critic network.

The control law is only updated when the triggering condition (13) is violated. Since the design of the neural-network-based observer can reconstruct both the system internal state and the control coefficient function, the control law can be directly calculated as

$$\mu(\hat{x}_j) = -\frac{1}{2}R^{-1}g^T(\hat{x}_j)\hat{V}_{\hat{x}_j} \tag{61}$$

where $\hat{V}_{\hat{x}_j}$ is the partial derivative of the estimated performance index with respect to the sampled state $\hat{x}_j$. According to (57), $\hat{V}_{\hat{x}_j}$ can be formulated as

$$\begin{aligned}\hat{V}_{\hat{x}_j} &= \frac{\partial\hat{V}(\hat{x}_j)}{\partial\hat{x}_j} \\ &= \frac{\partial\hat{V}(\hat{x}_j)}{\partial\Phi(m(\hat{x}_j))}\frac{\partial\Phi(m(\hat{x}_j))}{\partial m(x(\hat{x}_j))}\frac{\partial m(x(\hat{x}_j))}{\partial\hat{x}_j} \\ &= \frac{1}{2}\hat{\omega}_{c2j}^T\Big(1 - \Phi^2\big(m(\hat{x}_j)\big)\Big)\omega_{c1(\hat{x}_j)} \end{aligned} \tag{62}$$

Algorithm 1 Event-Triggered ADP Control Design Using Only the Measurable Input–Output Data

Set $i = 0$, $j = 0$, $\hat{x}_0 = x_0$
Calculate $\mu(\hat{x}_j) = -\frac{1}{2}R^{-1}g^T(\hat{x}_j)\hat{V}_{\hat{x}_j}$
Initialize all the neural network weights
for all $i < N_{run}$ **do**
  State estimation:
  $\dot{\hat{x}} = A\hat{x} + \hat{\omega}_{o2j}^T\Phi(\hat{x}_j) + G(y - C\hat{x})$
  Policy evaluation:
  $V(\hat{x}) = \min_{\mu(\hat{x}_j)} \int_0^\infty U(\hat{x}(\tau), \mu(\hat{x}_j))d\tau$
  if $\hat{x}_j - \hat{x} = \hat{e}_j > e_T$ **then**
    Set $j = j + 1$, $\hat{x}_j = \hat{x}$
    Update $\hat{\omega}_{o2j}$ according to (39)
    Update $\mu(\hat{x}_j) = \arg\min_{\mu(\hat{x}_j)}\{V(\hat{x}_j)\}$
  end if
  Update system information $\dot{x} = F(x, \mu(\hat{x}_j))$; $y = Cx$
  Set $i = i + 1$
end for

to which $\omega_{c1}(\hat{x}_j)$ is the fixed weights of $\hat{x}$ component for the input to the hidden layer of the critic network at the jump instant δ_j.

Also, considering (29), $g(\hat{x}_j)$ can be described by

$$\begin{aligned} g(\hat{x}_j) &= \frac{\partial\left(F_A(\hat{x}_j) + g(\hat{x}_j)\mu(\hat{x}_j)\right)}{\partial\mu(\hat{x}_j)} \\ &= \frac{\partial\left(F_A(\hat{x}_j) + g(\hat{x}_j)\mu(\hat{x}_j)\right)}{\partial\Phi(\hat{x}_j)}\frac{\partial\Phi(\hat{x}_j)}{\partial\mu(\hat{x}_j)} \\ &= \frac{1}{2}\hat{\omega}_{o2j}^T\left(1 - \Phi^2(\hat{x}_j)\right)\omega_{o1}\left(\mu(\hat{x}_j)\right) \end{aligned} \tag{63}$$

where $\omega_{o1}(\mu(\hat{x}_j))$ is the input-to-hidden layer weights of $\mu(\hat{x}_j)$ component for the function network at jump instant δ_j. Because the control law is only updated when the triggering condition (13) is violated, we then have the following description:

$$u(t) = \begin{cases} \mu(\hat{x}_{j-1}), & \text{Event is not triggered, } \delta_{j-1} \le t < \delta_j \\ -\frac{1}{2}R^{-1}g^T(\hat{x}_j)\hat{V}_{\hat{x}_j}, & \text{Event is triggered, } t = \delta_j. \end{cases} \tag{64}$$

The algorithm of the proposed event-triggered ADP control using the measurable input–output data is provided in Algorithm 1.

D. Stability Analysis of the Closed-Loop System

In this section, the stability analysis for the close-loop system will be investigated. A Lyapunov function candidate is considered as a combination of the Lyapunov functions for the neural-network-based observer and the designed control law. Both of them have two dynamics. The following theorem provides the stability of the whole system.

Theorem 3: Consider the nonlinear continuous-time system (1) with the event-triggered observer (32) and control law (64). The tuning laws for the impulsive observer and the continuous critic network are provided by (36), (39), and (60), respectively. Then, the system state x, sampled state $\hat{x}_j$, observer error $\tilde{x}$, function network weights estimation error $\tilde{\omega}_{o2}$, and the critic network weights estimation error $\tilde{\omega}_{c2}$ are all UUB given the following triggering condition:

$$\left\|e_{y_j}\right\|^2 \le \frac{\left(1 - \alpha^2\right)\underline{\lambda}(Q)\|C\|^2\|y\|^2 + \|C\|^2\|r^T\mu(x_j)\|^2}{L^2\|r\|^2} \tag{65}$$

where $\alpha \in (0, 1)$.

Proof: The proof of the boundedness is carried out in two parts, which are the continuous and the jump dynamics, respectively. The objective is to prove that both dynamics of the impulsive close-loop model are UUB. First, let us consider the following Lyapunov function:

$$\begin{aligned} L_{cl} &= \frac{1}{2}\tilde{x}^T P\tilde{x} + \frac{1}{2}tr\left(\tilde{\omega}_{o2}^T\tilde{\omega}_{o2}\right) + V^*(x) + V^*(\hat{x}_j) \\ &\quad + \frac{\beta_c^{-1}}{2}tr\left(\tilde{\omega}_{c2}^T\tilde{\omega}_{c2}\right) \\ &= L_o + L_c, \quad t \in \left(\delta_j, \delta_{j+1}\right] \end{aligned} \tag{66}$$

where

$$L_o = \frac{1}{2}\tilde{x}^T P\tilde{x} + \frac{1}{2}tr\left(\tilde{\omega}_{o2}^T\tilde{\omega}_{o2}\right) \tag{67}$$

$$L_c = V^*(x) + V^*(\hat{x}_j) + \frac{\beta_c^{-1}}{2}tr\left(\tilde{\omega}_{c2}^T\tilde{\omega}_{c2}\right) \tag{68}$$

and $V^*(x)$ and $V^*(\hat{x}_j)$ are the optimal performance index for the continuous and event-triggered sampled system.

For the continuous dynamics of the impulsive model, we take the time derivative of (66). $\dot{L}_o$ is provided in (43). Now $\dot{L}_c$ needs to be considered. Note that the second term in (68) has a zero derivative. Hence, we obtain

$$\dot{L}_c = \frac{\partial V^{*T}(x)}{\partial x}\dot{x} + \beta_c^{-1}tr\left(\tilde{\omega}_{c2}^T\dot{\tilde{\omega}}_{c2}\right) \tag{69}$$

where $\tilde{\omega}_{c2} = \omega_{c2}^* - \hat{\omega}_{c2}$, and

$$\begin{aligned} \dot{\tilde{\omega}}_{c2} &= \beta_c\frac{\kappa}{\left(\kappa^T\kappa + 1\right)^2}\left(\hat{\omega}_{c2}^T\kappa + U\left(\hat{x}, \mu(\hat{x}_j)\right)\right)^2 \\ &= -\beta_c\frac{\kappa\kappa^T}{\left(\kappa^T\kappa + 1\right)^2}\tilde{\omega}_{c2} \\ &\quad + \beta_c\frac{\kappa}{\left(\kappa^T\kappa + 1\right)^2}\left(\kappa^T\omega_{c2}^* + U\left(\hat{x}, \mu(\hat{x}_j)\right)\right) \\ &= -\beta_c\frac{\kappa\kappa^T}{\left(\kappa^T\kappa + 1\right)^2}\tilde{\omega}_{c2} + \beta_c\frac{\kappa}{\left(\kappa^T\kappa + 1\right)^2}\sigma_c \end{aligned} \tag{70}$$

where $\sigma_c = -(\partial\varepsilon_c/\partial\hat{x})\dot{\hat{x}}$.

Now, we will consider the following two terms separately:

$$\dot{L}_{c1}\left(V^*\right) = \frac{\partial V^{*T}(x)}{\partial x}\dot{x} \tag{71}$$

$$\dot{L}_{c2}(\tilde{\omega}_{c2}) = \beta_c^{-1}tr\left(\tilde{\omega}_{c2}^T\dot{\tilde{\omega}}_{c2}\right). \tag{72}$$

Then, (71) can be rewritten as

$$\begin{aligned} \dot{L}_{c1}\left(V^*\right) &= \frac{\partial V^{*T}(x)}{\partial x}\left(f(x) + g(x)\mu(\hat{x}_j)\right) \\ &= \frac{\partial V^{*T}(x)}{\partial x}f(x) + \frac{\partial V^{*T}(x)}{\partial x}g(x)\mu(\hat{x}_j). \end{aligned} \tag{73}$$

Consider (17) and (18), we obtain

$$\dot{L}_{c1}\left(V^*\right) = \frac{1}{4}V_x^{*T}g(x)R^{-1}g^T(x)V_x^* - y^TQy - 2\mu^{*T}(x)R\mu^*\left(\hat{x}_j\right) = \mu^{*T}(x)R\mu^*(x) - 2\mu^{*T}(x)R\mu^*\left(\hat{x}_j\right) - y^TQy \tag{74}$$

where $R = r \cdot r^T$ is a symmetric positive definite matrix. Therefore, we have

$$\mu^{*T}(x)R\mu^*(x) - 2\mu^{*T}(x)R\mu^*\left(\hat{x}_j\right) = \left\|r^T\mu^*(x) - r^T\mu^*\left(\hat{x}_j\right)\right\|^2 - \left\|r^T\mu^*\left(\hat{x}_j\right)\right\|^2. \tag{75}$$

By using the Lipschitz condition in Assumption 2, we have

$$\begin{aligned}\dot{L}_{c1}\left(V^*\right) &\le -\left\|r^T\mu^*\left(\hat{x}_j\right)\right\|^2 + L^2\|r\|^2\left\|e_{x_j}\right\|^2 - \underline{\lambda}(Q)\|y\|^2 \\ &= -\alpha^2\underline{\lambda}(Q)\|y\|^2 + \Big[-\left(1-\alpha^2\right)\underline{\lambda}(Q)\|y\|^2 \\ &\quad + L^2\|r\|^2\left\|e_{x_j}\right\|^2 - \left\|r^T\mu^*\left(\hat{x}_j\right)\right\|^2\Big].\end{aligned} \tag{76}$$

Considering the triggering condition (65), we have

$$\dot{L}_{c1}\left(V^*\right) \le -\alpha^2\underline{\lambda}(Q)\|y\|^2. \tag{77}$$

Next, for the term $\dot{L}_{c2}(\tilde{\omega}_{c2})$ in (72), we obtain

$$\begin{aligned}\dot{L}_{c2}(\tilde{\omega}_{c2}) &= \beta_c^{-1}tr\left(-\beta_c\tilde{\omega}_{c2}^T\frac{\kappa\kappa^T}{\left(\kappa^T\kappa+1\right)^2}\tilde{\omega}_{c2} + \beta_c\tilde{\omega}_{c2}^T\frac{\kappa}{\left(\kappa^T\kappa+1\right)^2}\sigma_c\right) \\ &\le -\left\|\frac{\kappa\kappa^T}{\kappa^T\kappa+1}\right\|^2\|\tilde{\omega}_{o2}\|^2 + \frac{1}{2\beta_c}\left(\beta_c^2\left\|\frac{\kappa\kappa^T}{\kappa^T\kappa+1}\right\|^2\|\tilde{\omega}_{o2}\|^2 + \frac{\sigma_c^2}{\left\|\kappa^T\kappa+1\right\|^2}\right) \\ &\le -\left(1-\frac{\beta_c}{2}\right)\left\|\frac{\kappa\kappa^T}{\kappa^T\kappa+1}\right\|^2\|\tilde{\omega}_{o2}\|^2 + \frac{\sigma_c^2}{2\beta_c}.\end{aligned} \tag{78}$$

It is important to note that the gradients of the critic network error is upper bounded, i.e., $\sigma_c \le \sigma_{cM}$. Hence, we have

$$\dot{L}_{c2}(\tilde{\omega}_{c2}) \le -\left(1-\frac{\beta_c}{2}\right)\left\|\frac{\kappa\kappa^T}{\kappa^T\kappa+1}\right\|^2\|\tilde{\omega}_{o2}\|^2 + \frac{\sigma_{cM}^2}{2\beta_c}. \tag{79}$$

Based on (43), (77), and (79), then $\dot{L}_{cl}$ becomes

$$\begin{aligned}\dot{L}_{cl} \le &-\frac{1}{2}\underline{\lambda}(M)\|\tilde{x}\|^2 + (2\omega_{oM}\Phi_M\|P\| + \|P\|\xi_M)\|\tilde{x}\| \\ &- \alpha^2\underline{\lambda}(Q)\|y\|^2 - \left(1-\frac{\beta_c}{2}\right)\left\|\frac{\kappa\kappa^T}{\kappa^T\kappa+1}\right\|^2\|\tilde{\omega}_{o2}\|^2 \\ &+ \frac{\sigma_{cM}^2}{2\beta_c}.\end{aligned} \tag{80}$$

Therefore, if the following conditions are satisfied:

$$\beta_c < 2 \tag{81}$$

$$\|\tilde{x}\| \ge \frac{4\omega_M\Phi_M\|P\| + 2\xi_M\|P\|}{\underline{\lambda}(M)} \tag{82}$$

$$\|\tilde{\omega}_{o2}\| \ge \sqrt{\frac{\sigma_{cM}^2/2\beta_c}{\left(1-\frac{\beta_c}{2}\right)\left\|\frac{\kappa\kappa^T}{\kappa^T\kappa+1}\right\|^2}} \tag{83}$$

then $\dot{L}_{cl} < 0$. This means the continuous dynamics of the impulsive model are UUB.

Now, we will consider the boundedness of the jump dynamics. The first difference of the Lyapunov function is shown as follows:

$$\begin{aligned}\Delta L_{cl} = &V^*\left(x^+\right) - V^*(x) + V^*\left(\hat{x}_j^+\right) - V^*\left(\hat{x}_j\right) \\ &+ \beta_c^{-1}tr\left(\left(\tilde{\omega}_{c2}^+\right)^T\tilde{\omega}_{c2}^+\right) - \beta_c^{-1}tr\left(\tilde{\omega}_{c2}^T\tilde{\omega}_{c2}\right) \\ &+ \frac{1}{2}\left(\tilde{x}^T\right)^+P\tilde{x}^+ - \frac{1}{2}\tilde{x}^TP\tilde{x} \\ &+ \frac{1}{2}tr\left(\left(\tilde{\omega}_{o2}^T\right)^+\tilde{\omega}_{o2}^+\right) - \frac{1}{2}tr\left(\tilde{\omega}_{o2}^T\tilde{\omega}_{o2}\right) \\ = &\Delta L_c + \Delta L_o, \quad t = \delta_j\end{aligned} \tag{84}$$

where ΔL_o is defined in (45), which is UUB under the conditions (54) and (55). Now, we consider the boundedness of ΔL_c which is defined as

$$\begin{aligned}\Delta L_c = &V^*\left(x^+\right) - V^*(x) + V^*\left(\hat{x}_j^+\right) - V^*\left(\hat{x}_j\right) \\ &+ \beta_c^{-1}tr\left(\left(\tilde{\omega}_{c2}^+\right)^T\tilde{\omega}_{c2}^+\right) - \beta_c^{-1}tr\left(\tilde{\omega}_{c2}^T\tilde{\omega}_{c2}\right).\end{aligned} \tag{85}$$

Since the states and the critic network estimation error are UUB from the first part of the proof, there exists $V^*(x^+) \le V^*(x)$ and $tr((\tilde{\omega}_{c2}^+)^T\tilde{\omega}_{c2}^+) \le tr(\tilde{\omega}_{c2}^T\tilde{\omega}_{c2})$ at the jump instants $t = \delta_j$. Moreover, for the sampled data, because during the jump instants, one has $\hat{x}^+ = \hat{x}_j^+$ and we have proved that the state estimation error is UUB, then $V^*(\hat{x}_j^+) \le V^*(\hat{x}_j)$. Therefore, we have $\Delta L_c < 0$, then $\Delta L_{cl} < 0$. This means the jump dynamics of the close-loop system is also UUB. This completes the proof. ■

IV. SIMULATION RESULTS

Consider a single link robot arm system giving by

$$\ddot{\theta}(t) = -\frac{MgH}{G}\sin(\theta(t)) - \frac{D}{G}\dot{\theta}(t) + \frac{1}{G}u(t) \tag{86}$$

where

$g = 9.81$	is the acceleration of gravity;
$H = 0.5$	is the length of the arm;
$D = 2$	is the viscous friction;
$M = 10$	is the mass of the payload;
$G = 10$	is the moment of inertia;
$\theta(t)$	is the angle position of robot arm;
$u(t)$	is the control input.

We assume that only the angle position $\theta(t)$ of the robot arm is observable. Defining $x_1(t) = \theta(t)$ and $x_2(t) = \dot{\theta}(t)$, the dynamic function (86) can be described as

$$\begin{cases}\dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{2}{10}x_2 + \frac{1}{10}u - \frac{49.05\sin(x_1)}{10} \\ y = x_1.\end{cases} \tag{87}$$

We can clearly observe that $y = x_1$ is the system measurable feedback in (87). This means the output matrix is $C = [1, 0]$ in this case.

We use the proposed event-triggered ADP method to solve the problem. In order to recover the internal system state, an

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

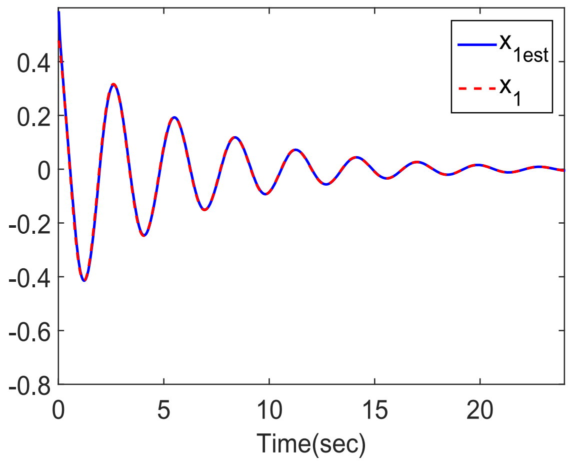


Fig. 2. System responses ($x_1/\hat{x}_1$) with the event-triggered observer and ADP controller.

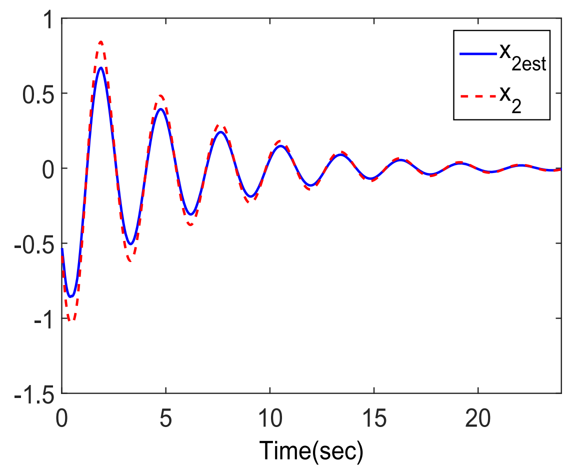


Fig. 3. System responses ($x_2/\hat{x}_2$) with the event-triggered observer and ADP controller.

observer is built with the following parameters:

$$A = \begin{bmatrix} 0 & 1 \\ -4 & -0.4 \end{bmatrix}; \qquad G = \begin{bmatrix} 10 & -1 \end{bmatrix}^T. \tag{88}$$

The designed observer includes a three-layer function network with the neuron structure as 3–6–2 (i.e., three input neurons, six hidden neurons, and two output neurons). Based on the estimated internal state from the observer, a critic network is established to approximate the performance index and help to obtain the control law. The neuron structure of the critic network is 3–8–1.

Choose the triggering condition as (13) with $L = 3$ and $\alpha = 0.95$. Set Q, r as the identity matrix with appropriate dimensions. Therefore, we have the triggering condition for this case as

$$\|e_{y_j}\|^2 \leq \frac{(1-0.95^2)\|C\|^2\|y\|^2 + \|C\|^2 \left\| r^T \mu(x_j) \right\|^2}{3^2}. \tag{89}$$

The trigger instants are decided according to (89). When the gap $e_{y_j} = C\hat{x}_j - y$ violates condition (89), the system state is sampled again by setting $\hat{x}_j = \hat{x}(t)$. The event-triggered observer and control law are updated again according to the sampled state.

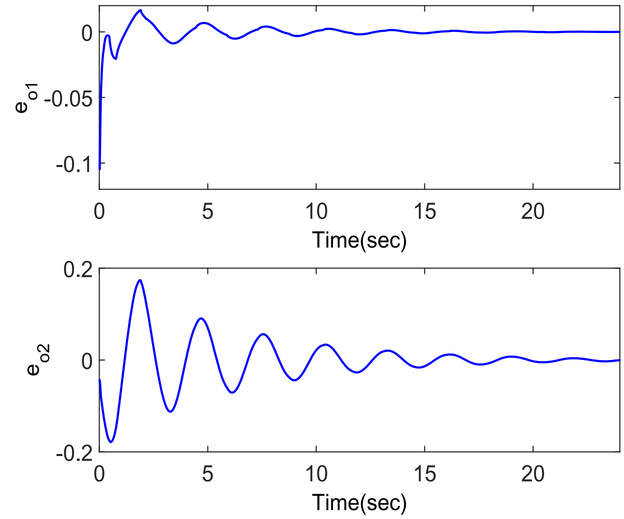


Fig. 4. Errors between the estimated state and the true state.

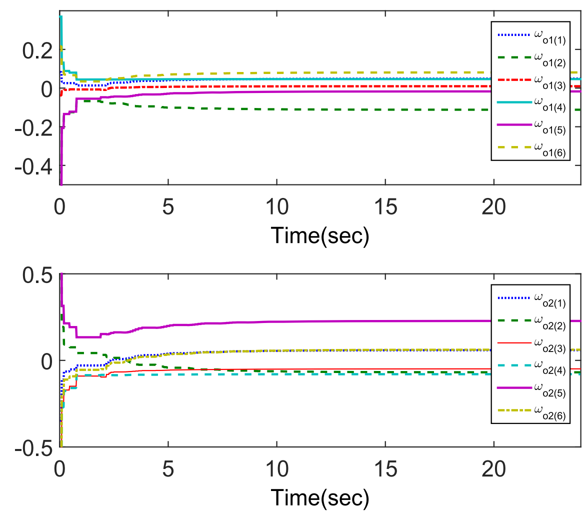


Fig. 5. Trajectory of the weights in function network.

Set the initial learning rates for both the function and the critic network as $\beta_o = \beta_c = 0.1$. Learning rates are decreased by 0.05 every five time steps until they reach $\beta_o = \beta_c = 0.005$ and stay thereafter. The initial weights of both networks are chosen randomly within $[-1, 1]$. The initial state is set to $x_0 = [0.5, -0.5]^T$. The sampling period for discretization is chosen as 0.03s.

By employing the event-triggered ADP control method proposed in this paper, we stabilize the partially observable system (87) only using the system input–output data. The trajectories of the system estimated state and true state are provided in Figs. 2 and 3. It can be seen that the estimated state $\hat{x}_1$ and $\hat{x}_2$ can quickly approach the true state x_1 and x_2, respectively. This means the designed observer can recover the system internal state from the output feedback, even with the reduced sampled data. The errors between the estimated state and the true state are provided in Fig. 4. The learning process of the function network weights are shown in Fig. 5. It is clearly that the weights updating law is aperiodic and only based on the sampled data. The observer is online training. The trajectory for the event-triggered control law in this

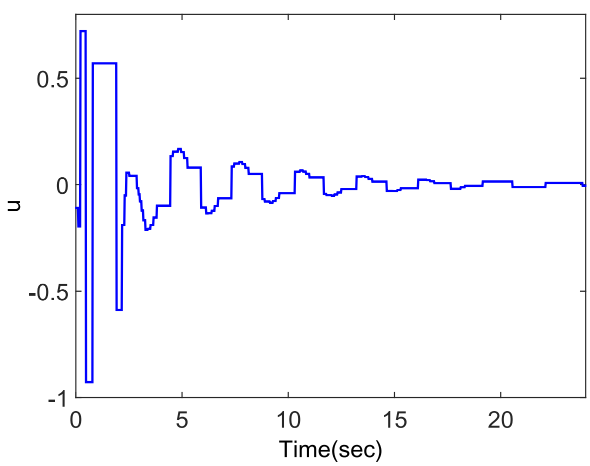


Fig. 6. Trajectory of the event-triggered control law.

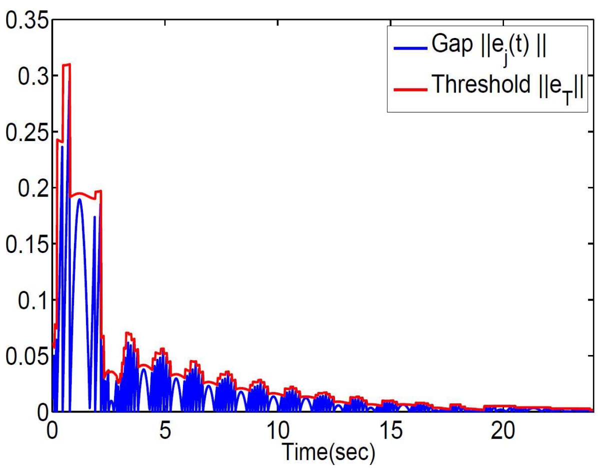


Fig. 7. Comparison of the gap $\|e_{y_j}\|$ and the threshold $\|e_T\|$.

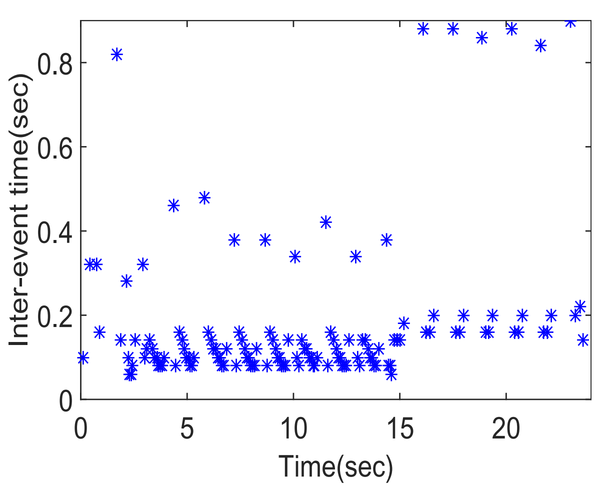


Fig. 8. Inter-event time during the learning process.

process is shown in Fig. 6. We can observe that the control law is a piecewise signal. This means the control law keeps the same at period $[\delta_j, \delta_{j+1})$ and is only updated when an event is triggered. The relationship between the gap $\|e_{y_j}\|$ and the threshold is shown in Fig. 7. It can be clearly observed that the gap $\|e_{y_j}\|$ is always smaller than the threshold to

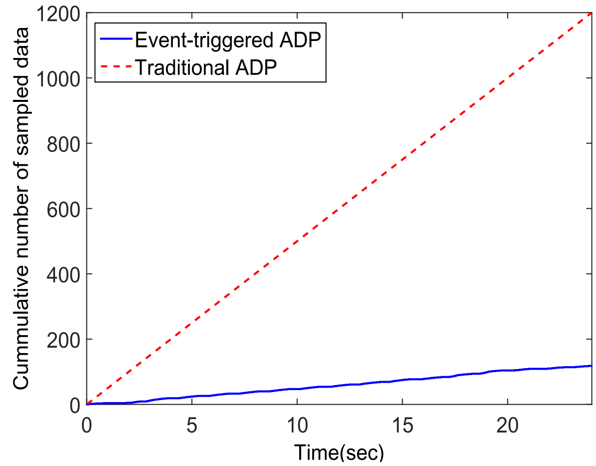


Fig. 9. Cumulative number of the sampled data for both the event-triggered ADP method and traditional ADP method.

make sure the close-loop system is stable. The inter-event time between two consecutive transmissions is shown in Fig. 8. We know the inter-event time exists and is up to 0.9 s in this case. Finally, the cumulative number of the sampled data during the control process for both the proposed event-triggered ADP method and the traditional ADP method in [40] are provided in Fig. 9. The event-triggered ADP method uses 118 samples while the traditional ADP method needs 1200 sample data. This means by efficiently reducing the sampled instants, the performance of the control method will not be influenced.

V. Conclusion

An event-triggered ADP control method was proposed in this paper for nonlinear continuous-time system using only the input–output data. A neural-network-based observer was established to reconstruct the system internal states and the control coefficient function. Neural network techniques were applied to approximate the performance index and help calculate the control law. In order to save the computation resource and transmission load, the designed observer and the controller were only updated when an event was triggered. The stability of the close-loop system was analyzed by Lyapunov construct for both the continuous and the jump dynamics. The simulation results demonstrated the effectiveness of the proposed method and also verified the theoretical analysis.

References

[1] J. Si, A. G. Barto, W. B. Powell, and D. C. Wunsch, Eds., *Handbook of Learning and Approximate Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2004.

[2] V. A. Ugrinovskii and H. R. Pota, "Decentralized control of power systems via robust control of uncertain Markov jump parameter systems," *Int. J. Control*, vol. 78, no. 9, pp. 662–677, 2005.

[3] S. C. Lee, "Maintenance strategies for manufacturing systems using Markov models," Ph.D. thesis, Dept. Mech. Eng., Univ. Michigan, Ann Arbor, MI, USA, 2010.

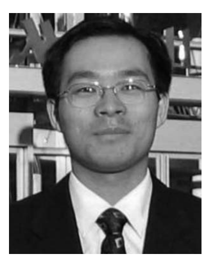
[4] H. Zhang, C. Qin, B. Jiang, and Y. Luo, "Online adaptive policy learning algorithm for state feedback control of unknown affine nonlinear discrete-time systems," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2706–2718, Dec. 2014.

[5] F. L. Lewis and D. Liu, Eds., *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. New York, NY, USA: Wiley, 2013.

[6] H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive Dynamic Programming for Control: Algorithms and Stability*. London, U.K.: Springer, 2013.
[7] H. Zhang, Q. Wei, and D. Liu, "An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games," *Automatica*, vol. 47, no. 1, pp. 207–214, 2011.
[8] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 206–216, Feb. 2013.
[9] F. L. Lewis, D. Liu, and G. G. Lendaris, "Special issue on adaptive dynamic programming and reinforcement learning in feedback control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 896–897, Aug. 2008.
[10] P. J. Werbos, "ADP: The key direction for future research in intelligent control and understanding brain intelligence," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 898–900, Aug. 2008.
[11] G. G. Lendaris, "Higher level application of ADP: A next phase for the control field?" *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 901–912, Aug. 2008.
[12] H. Zhang, H. Liang, Z. Wang, and T. Feng, "Optimal output regulation for heterogeneous multiagent systems via adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, 2016, in press.
[13] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.
[14] R. Song, W. Xiao, H. Zhang, and C. Sun, "Adaptive dynamic programming for a class of complex-valued nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1733–1739, Sep. 2014.
[15] H. Zhang, C. Qin, and Y. Luo, "Neural-network-based constrained optimal control scheme for discrete-time switched nonlinear system using dual heuristic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 839–849, Jul. 2014.
[16] H. Xu, Q. Zhao, and S. Jagannathan, "Finite-horizon near-optimal output feedback neural network control of quantized nonlinear discrete-time systems with input constraint," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1776–1788, Aug. 2015.
[17] Y. Jiang and Z.-P. Jiang, "Robust adaptive dynamic programming and feedback stabilization of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 882–893, May 2014.
[18] Y. Fu, J. Fu, and T. Chai, "Robust adaptive dynamic programming of two-player zero-sum games for continuous-time linear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3314–3319, Dec. 2015.
[19] X. Zhong, H. He, and D. V. Prokhorov, "Robust controller design of continuous-time nonlinear system using neural network," in *Proc. Int. Joint Conf. Neural Netw.*, Dallas, TX, USA, 2013, pp. 1–8.
[20] X. Zhong, H. He, H. Zhang, and Z. Wang, "Optimal control for unknown discrete-time nonlinear Markov jump systems using adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2141–2155, Dec. 2014.
[21] Z. Ni, H. He, X. Zhong, and D. V. Prokhorov, "Model-free dual heuristic dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1834–1839, Aug. 2015.
[22] X. Zhong, Z. Ni, and H. He, "A theoretical foundation of goal representation heuristic dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, 2016, in press.
[23] M. Hauskrecht, "Value-function approximations for partially observable Markov decision processes," *J. Artif. Intell. Res.*, vol. 13, no. 1, pp. 33–94, Aug. 2000.
[24] T. Smith and R. Simmons, "Heuristic search value iteration for POMDPs," in *Proc. 20th Conf. Uncertainty Artif. Intell.*, Banff, AB, Canada, 2004, pp. 520–527.
[25] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *Proc. IJCAI*, vol. 3. Acapulco, Mexico, 2003, pp. 1025–1032.
[26] H. Zhang, "Partially observable Markov decision processes: A geometric technique and analysis," *Oper. Res.*, vol. 58, no. 1, pp. 214–228, 2010.
[27] T. Jaakkola, S. P. Singh, and M. I. Jordan, "Reinforcement learning algorithm for partially observable Markov decision problems," in *Proc. NIPS*, vol. 7. Denver, CO, USA, 1995, pp. 345–352.
[28] E. Saad, "Reinforcement learning in partially observable Markov decision processes using hybrid probabilistic logic programs," *arXiv e-prints arXiv:1011.5951*, Nov. 2010.
[29] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.
[30] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input–output measured data," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2770–2779, Dec. 2015.
[31] Z. Ni, H. He, and X. Zhong, "Experimental studies on data-driven heuristic dynamic programming for POMDP," in *Frontiers of Intelligent Control and Information Processing*. Singapore: World Sci., 2015.
[32] X. Zhong, Z. Ni, Y. Tang, and H. He, "Data-driven partially observable dynamic processes using adaptive dynamic programming," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*, Orlando, FL, USA, 2014, pp. 1–8.
[33] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 847–861, Apr. 2013.
[34] P. Tallapragada and N. Chopra, "On event triggered tracking for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2343–2348, Sep. 2013.
[35] M. Lemmon, "Event-triggered feedback in control, estimation, and optimization," in *Networked Control Systems*. London, U.K.: Springer, 2010, pp. 293–358.
[36] J. Zhang and G. Feng, "Event-driven observer-based output feedback control for linear systems," *Automatica*, vol. 50, no. 7, pp. 1852–1859, 2014.
[37] K. G. Vamvoudakis, "Event-triggered optimal adaptive control algorithm for continuous-time nonlinear systems," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 3, pp. 282–293, Jul. 2014.
[38] A. Sahoo, H. Xu, and S. Jagannathan, "Near optimal event-triggered control of nonlinear discrete-time systems using neurodynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, 2016, in press.
[39] A. Sahoo, H. Xu, and S. Jagannathan, "Neural network-based event-triggered state feedback control of nonlinear continuous-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, 2016, in press.
[40] D. Liu, Y. Huang, D. Wang, and Q. Wei, "Neural-network-observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming," *Int. J. Control*, vol. 86, no. 9, pp. 1554–1566, 2013.
[41] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.
[42] H. A. Talebi, F. Abdollahi, R. V. Patel, and K. Khorasani, *Neural Network-Based State Estimation of Nonlinear Systems*. New York, NY, USA: Springer, 2010.

Xiangnan Zhong received the B.S. degree in automation and the M.S. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2010 and 2012, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA.

Her current research interests include adaptive dynamic programming, reinforcement learning, neural network, and optimal control.



Haibo He (SM'11) received the B.S. and M.S. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University, Athens, OH, USA, in 2006.

From 2006 to 2009, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA. He is currently the Robert Haas Endowed Chair Professor with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA. He has published one sole-author research book (Wiley), edited one book (Wiley–IEEE), and six conference proceedings (Springer), and authored and co-authored over 200 peer-reviewed journal and conference papers. His current research interests include adaptive dynamic programming, computational intelligence, self-adaptive systems, machine learning, and data mining.

Prof. He was a recipient of the National Science Foundation CAREER Award in 2011 and the Providence Business News "Rising Star Innovator of The Year" Award in 2011. He severed as the General Chair of the IEEE Symposium Series on Computational Intelligence. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.